

**IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD**

Publication number: JP2006515941 (T)

Publication date: 2006-06-08

Inventor(s):

Applicant(s):

Classification:






- International: G06F15/80; G06F9/30; G06F9/38; G06F9/445; G06F9/46; G06F15/78; G06F15/76; G06F9/30; G06F9/38; G06F9/445; G06F9/46

- European: G06F9/3884

Application number: JP2005050222T 20031031

Priority number(s): US20030422503P 20021031; US20030683929 20031009; US20030683932 20031009; US20030684053 20031009; US20030684057 20031009; US20030684102 20031009; WO2003063455 20031031

Also published as:

 WO2004042560 (A2) WO2004042560 (A3) WO2004042574 (A2) WO2004042574 (A3) WO2004042569 (A2)

more &gt;&gt;

Abstract not available for JP 2006515941 (T)

Abstract of corresponding document: **WO 2004042560 (A2)**

A peer-vector machine includes a host processor and a hardwired pipeline accelerator. The host processor executes a program, and, in response to the program, generates host data, and the pipeline accelerator generates pipeline data from the host data. Alternatively, the pipeline accelerator generates the pipeline data, and the host processor generates the host data from the pipeline data. Because the peer-vector machine includes both a processor and a pipeline accelerator, it can often process data more efficiently than a machine that includes only processors or only accelerators. For example, one can design the peer-vector machine so that the host processor performs decision-making and non-mathematically intensive operations and the accelerator performs non-decision-making and mathematically intensive operations. By shifting the mathematically intensive operations to the accelerator, the peer-vector machine often can, for a given clock frequency, process data at a speed that surpasses the speed at which a processor-only machine can process the data.

Data supplied from the **esp@cenet** database — Worldwide

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2006-515941

(P2006-515941A)

(43) 公表日 平成15年6月8日(2006. 6. 8)

(51) Int. Cl.

G06F 15/80 (2006.01)

F 1

G06F 15/80

テーマコード (参考)

審査請求 未請求 予備審査請求 未請求 (全 27 頁)

(21) 出願番号	特願2005-502222 (P2005-502222)	(71) 出願人	504275476
(86) (22) 出願日	平成15年10月31日(2003. 10. 31)		ロックヒード マーティン コーポレーション
(85) 翻訳文提出日	平成17年6月16日(2005. 6. 16)		ン
(86) 国際出願番号	PCT/US2003/034555		アメリカ合衆国 バージニア州 2011
(87) 国際公開番号	*03004/042561		O, マナサッス, ゴッドウィン ドライブ
(87) 国際公開日	平成16年5月21日(2004. 5. 21)		9500, メール ドロップ 043,
(31) 優先権主張番号	60/422, 503		ビルディング 400
(32) 優先日	平成14年10月31日(2002. 10. 31)	(74) 代理人	100083832
(33) 優先権主張国	米国 (US)		弁理士 廣江 武典
(31) 優先権主張番号	10/683, 929	(74) 代理人	100129698
(32) 優先日	平成15年10月9日(2003. 10. 9)		弁理士 武川 隆寛
(33) 優先権主張国	米国 (US)	(74) 代理人	100129676
(31) 優先権主張番号	10/683, 932		弁理士 ▲高▼荒 新一
(32) 優先日	平成15年10月9日(2003. 10. 9)	(74) 代理人	100130074
(33) 優先権主張国	米国 (US)		弁理士 中村 繁元

最終頁に続く

(64) 【発明の名称】 多数パイプライン・ユニットを有するパイプライン加速器、関連計算マシン、並びに、方法

## (57) 【要約】

パイプライン加速器は、バスと複数のパイプライン・ユニットを含み、各ユニットがバスと結合されていると共に少なくとも1つの対応するハードウェアに組み込まれたパイプライン回路を含む。複数のパイプライン・ユニットをパイプライン加速器に含ませることによって、単一のパイプライン・ユニット加速器と比較して加速器のデータ処理性能を増大することができる。更には、複数のパイプライン・ユニットが共通バスを介して通信するようにそれらパイプライン・ユニットを設計することによって、パイプライン・ユニットの数を減らすことができ、よって、単にそれらパイプライン・ユニットをバスと結合するか或は該バスから結合解除することによって加速器のコンフィギュレーション及び機能を変えることができる。これは、それらパイプライン・ユニットの内の1つを変えるたび、或は、加速器内のパイプライン・ユニットの数を減らすたびに、それらパイプライン・ユニット・インターフェースを設計或は再設計する必要性をなくする。

10

## 【特許請求の範囲】

## 【請求項1】

パイプライン加速器であって、  
通信バスと、  
各々が、前記通信バスと結合されると共に各ハードウェアに組み込まれたパイプライン回路を含む複数のパイプライン・ユニットと、  
を備えるパイプライン加速器。

## 【請求項2】

前記パイプライン・ユニットの各々が、  
前記ハードウェアに組み込まれたパイプライン回路と結合された各メモリを備え、  
前記ハードウェアに組み込まれたパイプライン回路が、  
前記通信バスからデータを受信し、  
前記データを前記メモリにロードし、  
前記メモリから前記データを検索し、  
前記検索データを処理し、  
前記処理データを前記通信バスに駆動するように動作可能である、  
請求項1に記載のパイプライン加速器。

## 【請求項3】

前記パイプライン・ユニットの各々が、  
前記ハードウェアに組み込まれたパイプライン回路と結合された各メモリを備え、  
前記ハードウェアに組み込まれたパイプライン回路が、  
前記通信バスからデータを受信し、  
前記データを処理し、  
前記処理データを前記メモリにロードし、  
前記処理データを前記メモリから検索し、  
前記検索データを前記通信バスにロードすることができる、  
請求項1に記載のパイプライン加速器。

## 【請求項4】

前記ハードウェアに組み込まれたパイプライン回路の各々が各フィールド・プログラマブル・ゲート・アレイ上に配置されている、請求項1に記載のパイプライン加速器。

## 【請求項5】

パイプライン・バスと、  
前記通信バスと前記パイプライン・バスとに結合されたパイプライン・バス・インターフェースとを更に備える、請求項1に記載のパイプライン加速器。

## 【請求項6】

前記通信バスが、各々が各パイプライン・ユニットと結合されている複数の分岐を含むことと、  
前記分岐の各々と結合されたルータと、  
を更に含む、請求項1に記載のパイプライン加速器。

## 【請求項7】

前記通信バスが、各々が各パイプライン・ユニットと結合されている複数の分岐を含むことと、  
前記分岐の各々と結合されたルータと、  
パイプライン・バスと、  
前記ルータと前記パイプライン・バスとに結合されたパイプライン・バス・インターフェースと、  
を更に含む、請求項1に記載のパイプライン加速器。

## 【請求項8】

前記通信バスが、各々が各パイプライン・ユニットと結合されている複数の分岐を含むことと、

前記分岐の各々と結合されたルータと、  
パイプライン・バスと、  
前記ルータと前記パイプライン・バスとに結合されたパイプライン・バス・インターフェースと、  
前記ルータと結合された第2バスと、  
を更に含む、請求項1に記載のパイプライン加速器。

【請求項9】

前記通信バスが前記パイプライン・ユニットの内の1つにアドレスされたデータを受信するように動作でき、  
前記1つのパイプライン回路が前記データを受領するように動作でき、  
前記他のパイプライン回路が前記データを拒絶するように動作できる、請求項1に記載のパイプライン加速器。

10

【請求項10】

前記通信バスが、各々が各パイプライン・ユニットと結合されている複数の分岐を含むことと、  
前記分岐の各々と結合されたルータであり、  
前記パイプライン・ユニットの内の1つにアドレスされたデータを受信し、  
前記通信バスの前記各分岐を介して前記1つのパイプライン・ユニットに前記データを提供するように動作できるルータと、  
を含む、請求項1に記載のパイプライン加速器。

20

【請求項11】

計算マシンであって、  
プロセッサと、  
前記プロセッサと結合されたパイプライン・バスと、  
パイプライン加速器であり、  
通信バスと、  
前記パイプライン・バス及び前記通信バスの間に結合されたパイプライン・バス・インターフェースと、  
各々が前記通信バスと結合されると共に各ハードウェアに組み込まれたパイプライン回路を備える複数のパイプライン・ユニットと、を含むことから成るパイプライン加速器と、  
を備える計算マシン。

30

【請求項12】

前記プロセッサが前記パイプライン・ユニットの内の1つを識別するメッセージを生成して、該メッセージを前記パイプライン・バスに駆動するように動作でき、  
前記パイプライン・バス・インターフェースが前記メッセージを前記通信バスに結合するように動作でき、  
前記パイプライン・ユニットの各々が前記メッセージを分析するように動作でき、  
前記識別されたパイプライン・ユニットが前記メッセージを受領するように動作でき、  
前記他のパイプライン回路が前記メッセージを拒絶するように動作できる、請求項11に記載の計算マシン。

40

【請求項13】

前記通信バスが、各々が各パイプライン・ユニットと結合されている複数の分岐を備えることと、  
前記プロセッサが前記パイプライン・ユニットの内の1つを識別するメッセージを生成して、該メッセージを前記パイプライン・バスに駆動するように動作できることと、  
前記分岐の各々と結合されると共に前記パイプライン・バス・インターフェースと結合されて、前記パイプライン・バス・インターフェースから前記メッセージを受信してそのメッセージを前記識別されたパイプライン・ユニットに提供するように動作できるルータと、

50

を更に含む、請求項 11 に記載の計算マシン。

【請求項 14】

前記通信バスが、各々が各パイプライン・ユニットと結合されている複数の分岐を備えることと、

第 2 バスと、

前記分岐の各々、前記パイプライン・バス・インターフェース、並びに、前記第 2 バスとそれぞれ結合されたルータと、

を更に含む、請求項 11 に記載の計算マシン。

【請求項 15】

方法であって、

通信バスを介して、各々が各ハードウェアに組み込まれたパイプラインを含む複数のパイプライン・ユニットの内の第 1 のものにデータを送信することと、

前記データを前記第 1 パイプライン・ユニットで処理することと、

を含む方法。

10

【請求項 16】

前記データを送信することが、

前記データをルータに送信することと、

前記通信バスの各第 1 分岐を介して、前記ルータによって前記第 1 パイプライン・ユニットに前記データを提供することと、

を含む、請求項 15 に記載の方法。

20

【請求項 17】

前記データを送信することが、プロセッサによって前記第 1 パイプライン・ユニットに前記データを送信することを含む、請求項 15 に記載の方法。

【請求項 18】

前記データを送信することが、前記複数のパイプライン・ユニットの第 2 のものによって前記第 1 パイプラインに前記データを送信することを含む、請求項 15 に記載の方法。

【請求項 19】

前記処理データを前記第 1 パイプライン・ユニットによって前記通信バスに駆動することを更に含む、請求項 15 に記載の方法。

【請求項 20】

30

前記データを前記第 1 パイプライン・ユニットによって処理することが、

ハードウェアに組み込まれたパイプライン回路によって前記通信バスから前記データを受信することと、

前記ハードウェアに組み込まれたパイプライン回路によって前記データをメモリにロードすることと、

前記ハードウェアに組み込まれたパイプライン回路によって前記メモリから前記データを検索することと、

前記ハードウェアに組み込まれたパイプライン回路によって前記検索データを処理することと、

含む、請求項 15 に記載の方法。

40

【請求項 21】

前記データを前記第 1 パイプライン・ユニットによって処理することが、

ハードウェアに組み込まれたパイプライン回路によって前記通信バスから前記データを受信することと、

前記ハードウェアに組み込まれたパイプライン回路によって前記受信データを処理することと、

前記ハードウェアに組み込まれたパイプライン回路によって前記処理データをメモリにロードすることと、

前記メモリから前記処理データを検索し、前記ハードウェアに組み込まれたパイプライン回路によって該処理データを前記通信バスに駆動することと、

50

含む、請求項 15 に記載の方法。

【請求項 22】

前記データを含み且つ前記第 1 バイプライン・ユニットをメッセージの受取人として識別するメッセージを生成することと、

前記データを前記第 1 バイプライン・ユニットに送信することが、前記メッセージから前記第 1 バイプラインが前記メッセージの受取人であることを決定することを含むことと

を更に含む、請求項 15 に記載の方法。

【発明の詳細な説明】

【技術分野】

10

【0001】

<優先権の請求>

この出願は、下記の特許文献 1 に対する優先権を請求するものであり、引用することによってここに合体させる。

【特許文献 1】米国仮出願第 60/422,503 号（2002 年 10 月 31 日出願）

【0002】

<関連出願の相互参照>

この出願は、「改善された計算アーキテクチャ、関連システム、並びに、方法」と題された下記の特許文献 2、「改善された計算アーキテクチャを有する計算マシン、関連システム、並びに、方法」と題された下記の特許文献 3、「改善された計算アーキテクチャ用 20  
バイプライン加速器、関連システム、並びに、方法」と題された下記の特許文献 4、「プログラマブル回路、関連計算マシン、並びに、方法」と題された下記の特許文献 5 と関連し、これら特許文献は全て 2003 年 10 月 9 日に出願され、共通の所有者を有し、引用することによってここに合体させる。

【特許文献 2】米国出願第 10/684,102 号

【特許文献 3】米国出願第 10/684,053 号

【特許文献 4】米国出願第 10/683,929 号

【特許文献 5】米国出願第 10/684,057 号

【背景技術】

【0003】

30

比較的大量のデータを比較的短い期間で処理する通常の計算アーキテクチャは、処理負担を分担する多数の相互接続プロセッサを含む。処理負担を分担することによって、これら多数のプロセッサは、しばしば、所与のクロック周波数で単一プロセッサができるものよりよりも迅速にデータを処理できる。例えば、これらプロセッサの各々はデータの各部分を処理できるか、或は、処理アルゴリズムの各部分を実行できる。

【0004】

図 1 は、多数プロセッサ・アーキテクチャを有する従来の計算マシン 10 の概略ブロック図である。この計算マシン 10 は、マスター・プロセッサ 12 と、相互に通信すると共に該マスター・プロセッサとバス 16 を介して通信する共同プロセッサ 14<sub>1</sub>〜14<sub>n</sub>と、遠隔装置（図 1 では不図示）から生データを受け取る入力ポート 18 と、該遠隔装置に処理データを提供する出力ポート 20 とを含む。また、計算マシン 10 はマスター・プロセッサ 12 に対するメモリ 22 と、共同プロセッサ 14<sub>1</sub>〜14<sub>n</sub>に対する各メモリ 24<sub>1</sub>〜24<sub>n</sub>と、マスター・プロセッサ及び共同プロセッサがバス 16 を介して共有するメモリ 26 とを含む。メモリ 22 はマスター・プロセッサ 12 に対するプログラム及び作業メモリの双方の役割を果たし、各メモリ 24<sub>1</sub>〜24<sub>n</sub>は各共同メモリ 14<sub>1</sub>〜14<sub>n</sub>に対するプログラム及び作業メモリの双方の役割を果たす。共有されたメモリ 26 は、マスター・プロセッサ 12 及び共同プロセッサ 14 がそれらの間でデータを転送すること、ポート 18 を介して遠隔装置からデータを転送すること、ポート 20 を介して遠隔装置にデータを転送することを可能としている。またマスター・プロセッサ 12 及び共同プロセッサ 14 は、マシン 10 が生データを処理する速度を制御する共通クロック信号を受け取る。 40  
50

## 【0005】

一般に、計算マシン10は、マスター・プロセッサ12及び共同プロセッサ14の間で生データの処理を効果的に分割する。ソナー・アレイ等の遠隔ソース（図1では不図示）は、ポート18を介して、生データに対する先入れ先出し（FIFO）バッファ（不図示）として作用する共有メモリ26の1つの区分に生データをロードする。マスター・プロセッサ12はバス16を介してメモリ26から生データを検索して、マスター・プロセッサ及び共同プロセッサ14はその生データを処理して、バス16を介して必要に応じてデータをそれらの間に転送する。マスター・プロセッサ12はその処理データを共有メモリ26内に規定された別のFIFOバッファ（不図示）にロードし、遠隔ソースがポート20を介してこのFIFOからその処理データを検索する。

10

## 【0006】

演算例において、計算マシン10は生データに対する $n+1$ 個の各演算を順次実行することによって該生データを処理し、これら演算は一体的に高速フーリエ変換（FFT）等の処理アルゴリズムを構成する。より詳細には、マシン10はマスター・プロセッサ12及び共同プロセッサ14からのデータ処理パイプラインを形成する。クロック信号の所与の周波数で、そうしたパイプラインはしばしばマシン10が単一プロセッサのみを有するマシンよりも高速に生データを処理することを可能としている。

## 【0007】

メモリ26内における生データFIFO（不図示）からの生データ検索後、マスター・プロセッサ12はその生データに対して三角関数等の第1番演算を実行する。この演算は第1番結果を生み出し、それをプロセッサ12がメモリ26内に規定された第1番結果FIFO（不図示）に記憶する。典型的には、プロセッサ12はメモリ22内に記憶されたプログラムを実行し、そのプログラムの制御の下で上述した動作を実行する。プロセッサ12はメモリ22を作業メモリとしても使用し得て、当該プロセッサが第1番演算の中間期間に生成するデータを一時的に記憶する。

20

## 【0008】

次に、メモリ26内における第1番結果FIFO（不図示）からの第1番結果検索後、共同プロセッサ14はその第1番結果に対して対数関数等の第2番演算を実行する。この第2番演算は第2番結果を生み出し、それを共同プロセッサ14がメモリ26内に規定された第2番結果FIFO（不図示）に記憶する。典型的には、共同プロセッサ14はメモリ24内に記憶されたプログラムを実行し、そのプログラムの制御の下で上述した動作を実行する。共同プロセッサ14はメモリ24を作業メモリとしても使用し得て、当該共同プロセッサが第2番演算の中間期間に生成するデータを一時的に記憶する。

30

## 【0009】

次に共同プロセッサ24<sub>1</sub>〜24<sub>n</sub>は、共同プロセッサ24<sub>1</sub>に対して先に議論されたものと同様に、（第2番結果〜第 $(n-1)$ 番）結果に対して（第3番演算〜第 $n$ 番）演算を順次実行する。

## 【0010】

共同プロセッサ24<sub>n</sub>によって実行される第 $n$ 番演算は最終結果、即ち処理データを生み出す。共同プロセッサ24<sub>n</sub>はその処理データをメモリ26内に規定された処理データFIFO（不図示）内にロードし、遠隔装置（図1では不図示）がこのFIFOからその処理データを検索する。

40

## 【0011】

マスター・プロセッサ12及び共同プロセッサ14は処理アルゴリズムの種々の演算を同時に実行するので、計算マシン10は、しばしば、種々の演算を順次実行する単一プロセッサを有する計算マシンよりも生データを高速に処理することができる。詳細には、単一プロセッサは、生データから成る先行集合に対する全 $(n+1)$ 個の演算を実行するまで、生データから成る新しい集合を検索できない。しかし、以上に議論したパイプライン技術を用いて、マスター・プロセッサ12は第1演算だけを実行後に生データから成る新しい集合を検索できる。結果として、所与のクロック周波数でこのパイプライン技術は、

50

単一プロセッサ・マシン（図1では不図示）と比較して約  $n+1$  倍だけマシン10が生データを処理する速度を増大することができる。

#### 【0012】

代替的には、計算マシン10は、生データに対するFFT等の処理アルゴリズムの  $(n+1)$  例を同時に実行することによって該生データを並列して処理し得る。即ち、もしそのアルゴリズムが先行する例において先に記載されたような  $(n+1)$  個の順次演算を含めば、マスター・プロセッサ12及び共同プロセッサ14の各々は生データからそれぞれが成る各集合に対して、順次、全  $(n+1)$  個の演算を実行する。その結果として、所与のクロック周波数で、先のパイプライン技術と同様のこの並列処理技術は、単一プロセッサ・マシン（図1では不図示）と比較して約  $n+1$  倍だけマシン10が生データを処理する速度を増大することができる。

#### 【0013】

残念ながら、計算マシン10は単一プロセッサ・計算マシン（図1では不図示）と比べてより迅速にデータを処理できるが、マシン10のデータ処理速度はしばしばプロセッサ・クロックの周波数より非常に小さい。詳細には、計算マシン10のデータ処理速度はマスター・プロセッサ12及び共同プロセッサ14がデータ処理するのに必要な時間によって制限される。簡略化のため、この速度制限の例はマスター・プロセッサ12と連携して議論されているが、この議論は共同プロセッサ14にも適用されることを理解して頂きたい。先に議論されたように、マスター・プロセッサ12は所望の方式でデータを操作すべくプロセッサを制御するプログラムを実行する。このプログラムはプロセッサ12が実行する複数の命令から成るシーケンスを含む。残念ながら、プロセッサ12は典型的には単一命令を実行するために多数のクロック・サイクルを必要とし、そしてしばしばデータの単一値を処理すべく多数の命令を実行しなければならない。例えば、プロセッサ12が第1データ値A（不図示）を第2データ値B（不図示）で乗算することを仮定する。第1クロック・サイクル中、プロセッサ12はメモリ22から乗算命令を検索する。第2及び第3クロック・サイクル中、プロセッサ12はメモリ26からA及びBをそれぞれ検索する。第4クロック・サイクル中、プロセッサ12はA及びBを乗算し、そして第5クロック・サイクル中に結果としての積をメモリ22或は26に記憶するか、或は、その結果としての積を遠隔装置（不図示）に提供する。これは最良ケースのシナリオであり、その理由は多くの場合にプロセッサ12はカウンタの初期化及び閉鎖等のオーバーヘッド・タスクに対して付加的なクロック・サイクルを必要とするからである。それ故に、よくてもプロセッサ12はA及びBを処理すべく5クロック・サイクルを必要とするか、或は、1データ値当たり平均2.5クロック・サイクルを必要とする。

#### 【0014】

結果として、計算マシン10がデータを処理する速度は、しばしば、マスター・プロセッサ12及び共同プロセッサ14を駆動するクロックの周波数より非常に低い。例えば、もしプロセッサ12は1.0ギガヘルツ（GHz）でクロックされるが、1データ値当たり平均2.5クロック・サイクルを必要とすれば、効果的なデータ処理速度は  $(1.0 \text{ GHz}) / 2.5 = 0.4 \text{ GHz}$  と同等である。この効果的なデータ処理速度は、しばしば、1秒当たり演算数の単位で特徴付けされる。それ故に、この例において、1.0 GHzのクロック速度で、プロセッサ12は0.4ギガ演算数/秒（Gops）で使用限界が定められる。

#### 【0015】

図2は、所与クロック周波数で且つしばしば該パイプラインがクロックされる速度と略同一速度で、プロセッサが可能であるよりは高速で典型的にはデータを処理できるハードウェアに組み込まれたデータ・パイプライン30のブロック線図である。パイプライン30は、プログラム命令を実行することなく、各データに対する各演算を各々が実行する演算子回路32、…、32を含む。即ち、所望の演算は回路32内に「書き込み」が為されて、それがプログラム命令の必要性なしに自動的にその演算を具現化するように為す。プログラム命令の実行と関連されたオーバーヘッドを減ずることによって、パイプライン30



は所与のクロック周波数でプロセッサが可能であるよりは単位秒当たりより多くの演算を典型的には実行する。

#### 【0016】

例えば、パイプライン30は所与のクロック周波数でプロセッサが可能であるよりは高速で以下の数式1をしばしば解くことができる。

$$Y(x_k) = (5x_k + 3)2^{x_k}$$

ここで、 $x_k$ は複数の生データ値から成るシーケンスを表す。この例において、演算子回路32<sub>1</sub>は $5x_k$ を計算する乗算器であり、回路32<sub>2</sub>は $5x_k + 3$ を計算する加算器であり、そして回路32<sub>3</sub> ( $n=3$ )は $(5x_k + 3)2^{x_k}$ を計算する乗算器である。

#### 【0017】

第1クロック・サイクル $k=1$ 中、回路32<sub>1</sub>はデータ値 $x_1$ を受け取って、それを5で乗じて、 $5x_1$ を生成する。

#### 【0018】

第2クロック・サイクル $k=2$ 中、回路32<sub>2</sub>は回路32<sub>1</sub>から $5x_1$ を受け取って、3を加えて、 $5x_1 + 3$ を生成する。またこの第2クロック・サイクル中に回路32<sub>1</sub>は $5x_2$ を生成する。

#### 【0019】

第3クロック・サイクル $k=3$ 中、回路32<sub>3</sub>は回路32<sub>2</sub>から $5x_1 + 3$ を受け取って、 $2^{x_1}$ で乗じて(効果としては、 $x_1$ だけ $5x_1 + 3$ を左シフトする)、第1結果 $(5x_1 + 3)2^{x_1}$ を生成する。またこの第3クロック・サイクル中に回路32<sub>1</sub>は $5x_3$ を生成し、回路32<sub>2</sub>は $5x_2 + 3$ を生成する。

#### 【0020】

このようにしてパイプライン30は、全ての生データ値が処理されるまで、引き続く生データ値 $x_k$ の処理を続行する。

#### 【0021】

結果として、生データ値 $x_1$ の受け取り後の2つのクロック・サイクルの遅延、即ち、この遅延はパイプライン30の待ち時間としばしば呼称され、パイプラインは結果 $(5x_1 + 3)2^{x_1}$ を生成し、その後、1つの結果を生成する、即ち各クロック・サイクル毎に $(5x_2 + 3)2^{x_2}$ 、 $(5x_3 + 3)2^{x_3}$ 、 $\dots$ 、 $(5x_n + 3)2^{x_n}$ を生成する。

#### 【0022】

待ち時間を無視して、パイプライン30はこうしてクロック速度と同等のデータ処理速度を有する。比較して、マスター・プロセッサ12及び共同プロセッサ14(図1)が先の例におけるようにクロック速度の0.4倍であるデータ処理速度を有すると仮定すれば、パイプライン30は、所与のクロック速度で、計算マシン10(図1)よりも2.5倍高速でデータを処理できる。

#### 【0023】

更に図2で参照されるように、設計者はフィールド・プログラマブル・ゲート・アレイ(FPGA)等のプログラマブル・ロジックIC(PLIC)にパイプライン30を具現化することを選ぶ可能性があり、その理由はPLICが特殊用途IC(ASIC)が為すよりも多くの設計及び変更の柔軟性を許容するからである。PLIC内にハードウェアに組み込まれた接続を構成するため、設計者はPLIC内に配置された相互接続構成レジスタを単に所定バイナリー状態に設定する。全てのこうしたバイナリー状態の組み合わせはしばしば「ファームウェア」と呼称される。典型的には、設計者はこのファームウェアをPLICと結合された不揮発性メモリ(図2では不図示)内にロードする。PLICを「ターンオン」すると、それはファームウェアをそのメモリから相互接続構成レジスタにダウンロードする。それ故に、PLICの機能を変更すべく、設計者は単にそのファームウェアを変更して、PLICがその変更されたファームウェアを相互接続構成レジスタにダウンロードすることを可能とする。ファームウェアを単に変更することによってPLICを変更する能力は、モデル作成段階中や「フィールド内」にパイプライン30をアップグレードするために特に有用である。

## 【0024】

残念ながら、ハードウェアに組み込まれたパイプライン30は重要な意思決定、特に入れ子意思決定を引き起こすアルゴリズムを実行すべき最良の選択でない可能性がある。プロセッサは、典型的には、入れ子意思決定命令（例えば、「もしAであれば、Bを為し、またもしCであれば、Dを為し、・・・、またnを為し等々」のように、入れ子条件命令）を、比肩する長さの演算命令（例えば、「 $A+B$ 」）を実行できる程に高速に実行できる。しかしパイプライン30は、比較的単純な決定（例えば、「 $A>B?$ 」）を効率的に為し得るが、典型的にはプロセッサができる程に効率的に入れ子決定（例えば、「もしAであれば、Bを為し、またもしCであれば、Dを為し、・・・またnを為す」）を為すことができない。この非効率性の1つの理由は、パイプライン30はほんの僅かなオンボード・メモリしか持たないことがあり、したがって外部作業/プログラム・メモリ（不図示）にアクセスすることを必要とすることがあるからである。そして、こうした入れ子決定を実行すべくパイプライン30を設計することができるが、必要とされる回路のサイズ及び複雑性はしばしばそうした設計を非現実的に為し、特にアルゴリズムが多数の種々の入れ子決定を含む場合にそうである。

10

## 【0025】

結果として、プロセッサは典型的には重要な意思決定を必要とする用途において使用され、ハードウェアに組み込まれたパイプラインは殆ど意思決定が為されないか或は意思決定されない「ナンバークランチング（数値データ処理）」用途に典型的には限定される。

## 【0026】

更には、下記に議論されるように、典型的には、特にパイプライン30が多数のPLICを含む場合、図2のパイプライン30等のハードウェアに組み込まれたパイプラインを設計/変更するよりも、図1の計算マシン10等のプロセッサに基づく計算マシンを設計/変更することが非常に易しい。

20

## 【0027】

プロセッサ及びそれらの周辺機器（例えば、メモリ）等の計算構成要素は、典型的には、プロセッサに基づく計算マシンを形成すべくそれら構成要素の相互接続を補助する工業規格通信インターフェースを含む。

## 【0028】

典型的には、規格通信インターフェースは2つの層、即ち、物理層及びサービス層を含む。

30

## 【0029】

物理層は、回路とこの回路のインターフェース及び動作パラメータを形成する対応回路相互接続とを含む。例えば、物理層はそれら構成要素を1つのバスに接続するピンと、それらのピンから受け取ったデータをラッチするバッファと、信号をそれらピンに駆動するドライバとを含む。動作パラメータは、ピンが受け取るデータ信号の許容可能電圧範囲と、データの書き込み及び読み取りのための信号タイミングと、動作の支援されたモード（例えば、バーストモード、ページモード）とを含む。従来の物理層はトランジスタ・トランジスタ論理（TTL）及びRAMBUSを含む。

## 【0030】

サービス層は、計算構成要素のデータ転送のためのプロトコルを含む。このプロトコルはデータのフォーマットと、構成要素によるフォーマット済みデータの送受信の方式とを含む。従来の通信プロトコルは、ファイル転送プロトコル（FTP）及び伝送制御プロトコル/インターネット・プロトコル（TCP/IP）を含む。

40

## 【0031】

結果として、製造業者やその他は工業規格通信インターフェースを有する計算構成要素を典型的には設定するので、そうした構成要素のインターフェースを典型的には設計でき、それを他の計算構成要素と比較的少ない労力で相互接続することができる。これは、計算マシンの他の部分の設計に設計者自信の時間を殆ど費やすことを可能として、各種構成要素を追加或は除去することによってそのマシンを変更することを可能としている。

50

## 【0032】

工業規格通信インターフェースを支援する計算構成要素を設計することは、設計ライブラリから既存の物理層を用いることによって設計時間を節約することを可能としている。これは、設計者が構成要素を既製の計算構成要素と容易にインターフェースすることを保証するものである。

## 【0033】

そして、共通した工業規格通信インターフェースを支援する計算構成要素を用いる計算マシンを設計することは、設計者がそれら構成要素を少しの時間及び労力で相互接続することを可能としている。それら構成要素は共通インターフェースを支援するので、設計者はそれらをシステム・バスを介して少しの設計労力で相互接続することができる。そして、その支援されたインターフェースは工業規格であるので、マシンを容易に変更することができる。例えば、システム設計が進化するに伴って種々の構成要素及び周辺機器をマシンに追加することができるか、或は、テクノロジーが進化するに伴って次世代の構成要素を追加／設計することが可能である。更には、構成要素が通常の工業規格サービス層を支援するので、計算マシンのソフトウェアに対応するプロトコルを具現化する既存のソフトウェア・モジュールを組み込むことができる。それ故に、インターフェース設計が本質的には既に整っているので少しの労力で構成要素をインターフェースでき、よって、マシンに所望の機能を実行させるマシンの各種部分（例えばソフトウェア）の設計に集中することができる。

## 【0034】

しかし残念ながら、図2のパイプライン30等のハードウェアに組み込まれたパイプラインを形成すべく、使用されるPLIC等の各種構成要素に対する既知の工業規格サービス層が全くない。

## 【0035】

結果として、多数のPLICを有するパイプラインを設計すべく、多大な時間を費やし、「ゼロから」設計し且つ種々のPLICの間の通信インターフェースのサービス層をデバッグする多大な労力を行行使す。典型的には、そうしたその場限りのサービス層は種々のPLIC間で転送されるデータのパラメータに依存する。同じように、プロセッサとインターフェースするパイプラインを設計すべく、パイプライン及びプロセッサの間の通信インターフェースのサービス層の設計及びデバッグに関して多大な時間を費やし且つ多大な労力を行行使す必要がある。

## 【0036】

同様に、そうしたパイプラインをPLICを該パイプラインに追加することによって変更すべく、典型的には、その追加されたPLICと既存のPLICとの間の通信インターフェースのサービス層の設計及びデバッグに関して多大な時間を費やし且つ多大な労力を行行使す。同じように、プロセッサを追加することによってパイプラインを変更すべく、或は、パイプラインを追加することによって計算マシンを変更すべく、パイプライン及びプロセッサの間の通信インターフェースのサービス層の設計及びデバッグに関して多大な時間を費やし且つ多大な労力を行行使しなけければならぬであろう。

## 【0037】

結果として、図1及び図2で参照されるように、多数のPLICをインターフェースすることとプロセッサをパイプラインにインターフェースすることとの難しさのため、計算マシンを設計する際に多大な妥協を為すことがしばしば強いられる。例えば、プロセッサに基づく計算マシンでは、ナンバークランピング速度を、複雑な意思決定を為す能力に対する設計／変更の柔軟性と交換することを強いられる。逆に、ハードウェアに組み込まれたパイプラインに基づく計算マシンでは、複雑な意思決定を為す能力と設計／変更の柔軟性を、ナンバークランピング速度と交換することを強いられる。更には、多数のPLICをインターフェースすることに関する難しさのため、少数のPLICよりも多くのPLICを有するパイプラインに基づくマシンを設計することはしばしば実際的ではない。その結果、実際的なパイプラインに基づくマシンはしばしば制限された機能しか有しない。そ

して、プロセッサをPLICとインターフェースすることに関する難しさのため、プロセッサを1つのPLICより多くのPLICにインターフェースすることは実際的ではない。その結果、プロセッサ及びパイプラインを組み合わせてことによって獲得される利益は最少となる。

【発明の開示】

【発明が解決しようとする課題】

【0038】

それ故に、プロセッサに基づくマシンの意思決定を為す能力を、ハードウェアに組み込まれたパイプラインに基づくマシンのナンバークランピング速度と組み合わせることを可能とする新しい計算アーキテクチャに対する要望が生じてきている。

10

【課題を解決するための手段】

【0039】

本発明の実施例に従えば、パイプライン加速器はバスと、各々がそのバスと結合されると共に少なくとも1つのハードウェアに組み込まれたパイプライン回路を含んでいる複数のパイプライン・ユニットを含む。

【0040】

複数のパイプライン・ユニットをパイプライン加速器に含ませることによって、単一パイプライン加速器と比較して加速器のデータ処理性能を増大することができる。更には、相互に通信すると共に他のピアと共通バスを介して通信するようにパイプライン・ユニットを設計することによって、パイプライン・ユニットの数を変えて、パイプラインのバスに対する単なる結合或は結合解除によって加速器のコンフィギュレーション及び機能を変えることができる。これは、パイプライン・ユニットの1つを変えるたび、或は、加速器内のパイプライン・ユニットの数を増減するたびに、パイプライン・ユニット・インターフェースを設計或は再設計することをなくするものである。

20

【発明を実施するための最良の形態】

【0041】

図3は、本発明の一実施例に従ったピア・ベクトル・アーキテクチャを有する計算マシン40の概略ブロック線図である。ホストプロセッサ42に加えて、ピア・ベクトル・マシン40はパイプライン加速器44を含み、それがデータ処理の少なくとも一部を実行して、図1の計算マシン10における共同プロセッサ14の列と効果的に置き換わる。それ故に、ホストプロセッサ42及び加速器44（又は以下に議論されるようにそのユニット）はデータ・ベクトルを前後に転送できる「ピア」である。加速器44はプログラム命令を実行しないので、所与のクロック周波数で共同プロセッサの列ができるものよりも著しく高速にデータに対して数学的に集約的な演算を典型的には実行する。結果として、プロセッサ42の意思決定能力と加速器44のナンバークランピング能力とを組み合わせることによって、マシン40はマシン10等の従来の計算マシンと同一の能力を有するが、しばしばそれよりもデータをより高速に処理することができる。更には、以下に議論されるように、加速器44にホストプロセッサ42の通信インターフェースと互換性がある通信インターフェースを設けることが、特にプロセッサの通信インターフェースが工業規格である場合に、マシン40の設計及び変更を補助する。そして、加速器44が多数のパイプライン・ユニット（例えば、PLICに基づく回路）を含む場合、それら各ユニットに同一の通信インターフェースを設けることが、特にそれら通信インターフェースが工業規格インターフェースと互換性がある場合に、当該加速器の設計及び変更を補助する。更には、マシン40は以下に議論されると共に先行して引用された特許出願におけるような他の長所等をも提供し得る。

30

40

【0042】

更に図3で参照されるように、ホストプロセッサ42及びパイプライン加速器44に加えて、ピア・ベクトル・計算マシン40は、プロセッサ・メモリ46、インターフェース・メモリ48、パイプライン・バス50、1つ或はそれ以上のファームウェア・メモリ52、任意選択的な生データ入力ポート54、処理済みデータ出力ポート58、並びに、任

50

意選択的なルータ 61 を含む。

【0043】

ホストプロセッサ 42 は処理ユニット 62 及びメッセージ・ハンドラー 64 を含み、プロセッサ・メモリ 46 は処理ユニット・メモリ 66 及びハンドラー・メモリ 68 を含み、そのそれぞれがプロセッサ・ユニット及びメッセージ・ハンドラーに対するプログラム及び作業の両メモリとして役立っている。プロセッサ・メモリ 46 は、加速器コンフィギュレーション・レジストリ 70 及びメッセージ・コンフィギュレーション・レジストリ 72 をも含み、それらが、ホストプロセッサ 42 が加速器 44 の機能を構成すると共に、該ホストプロセッサ 42 がメッセージ・ハンドラー 64 が送信及び受信するメッセージのフォーマットを構成することを可能とするそれぞれのコンフィギュレーション・データを記憶する。

10

【0044】

パイプライン加速器 44 は少なくとも 1 つの PLIC (図 4) 上に配置され、プログラム命令を実行することなしに各データを処理するハードウェアに組み込まれたパイプライン 74<sub>1</sub>、…74<sub>n</sub> を含む。ファームウェア・メモリ 52 は加速器 44 に対するコンフィギュレーション・ファームウェアを記憶する。もし加速器 44 が多数の PLIC 上に配置されたら、それら PLIC 及びそれらの各ファームウェア・メモリは多数パイプライン・ユニット内に配置され得るが、それらパイプラインは図 4 乃至図 8 と連携して以降に更に議論される。代替的には、加速器 44 は少なくとも 1 つの ASIC 上に配置され得る。その ASIC がひとたび形成されたならば構成不可能である内部相互接続を有し得る。この代替例において、マシン 40 はファームウェア・メモリ 52 を省略し得る。更には、加速器 44 が多数パイプライン 74<sub>1</sub>、…74<sub>n</sub> を含んで示されているが、ただ 1 つのパイプラインを含み得る。加えて、図示されていないが、加速器 44 はデジタル信号プロセッサ (DSP) 等の 1 つ或はそれ以上のプロセッサを含み得る。更には、図示されていないが、加速器 44 はデータ入力ポート及び/或はデータ出力ポートを含み得る。

20

【0045】

そしてホストプロセッサ 42 及びパイプライン加速器 44 は種々の IC 上に配置されているように議論されるが、ホストプロセッサ及びパイプライン加速器は同一の IC 上に配置され得る。

【0046】

ピア-ベクトル・マシン 40 の一般動作は、先行して引用された「改善された計算アーキテクチャ、関連システム、並びに、方法」と題された特許文献 2 に議論されており、ホストプロセッサ 42 の構造及び動作は、先行して引用された「改善された計算アーキテクチャを有する計算マシン、関連システム、並びに、方法」と題された特許文献 3 に議論されており、パイプライン加速器 44 の構造及び動作は、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献 4 と図 4 乃至図 8 と連携された以下に議論されている。

30

【0047】

図 4 は、本発明の一実施例に従った図 3 のパイプライン加速器 44 のユニット 78 のブロック線図である。

40

【0048】

加速器 44 は 1 つ或はそれ以上のそうしたパイプライン・ユニット 78 を含み (図 4 に 1 つのみ示される)、それらの各々は PLIC 或は ASIC 等のパイプライン回路 80 を含む。以下で更に議論されると共に先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献 4 にあるように、各パイプライン・ユニット 78 はホストプロセッサ 42 (図 3) の「ピア」であると共に加速器 44 の他のパイプライン・ユニットの「ピア」である。即ち、各パイプライン・ユニット 78 はホストプロセッサ 42 或は他の任意のパイプライン・ユニットと直接通信できる。よって、このピア-ベクトル・アーキテクチャは、もしパイプライン・ユニット 78 の全てがマスターパイプライン・ユニット (不図示) 或はホストプロセッサ 42

50

等の中央箇所を通じて通信した場合に生ずることとなるデータ「ボトルネック」を防止する。更にはこのアーキテクチャは、マシンに対する重大な変更なしに、ビタベクトル・マシン 40 (図 3) からピアを追加するか或はピアを除去することを可能とする。

#### 【0049】

パイプライン回路 80 は通信インターフェース 82 を含み、それが、ホストプロセッサ 42 (図 3) 等のピアと、通信シェル 84 を介したハードウェアに組み込まれたパイプライン 74<sub>1</sub>~74<sub>n</sub>、パイプライン・コントローラ 86、例外マネージャ 88、並びに、コンフィギュレーション・マネージャ 90 等の、パイプラインの他の構成要素との間でデータを転送する。パイプライン回路 80 は工業規格バス・インターフェース 91 及び通信バス 93 をも含み得て、インターフェース 82 をインターフェース 91 と接続する。代替的は、インターフェース 91 の機能は通信インターフェース 82 内に含まれ得て、バス 93 は省略される。

#### 【0050】

パイプライン回路 80 の複数の構成要素を複数の個別モジュールとして設計することによって、そのパイプライン回路の設計をしばしば簡略化することができる。即ち、それら構成要素の各々を個別に設計及び試験することができ、次いでそれらを統合するものであり、それはソフトウェア或はプロセッサに基づく計算システム (図 1 のシステム 10 等) を設計する際に行われることと非常に似ている。加えて、これら構成要素、特に設計者が他のパイプライン設計においてたぶん頻繁に使用するような通信インターフェース 82 等の構成要素を規定するハードウェア記述言語 (HDL) をライブラリ (不図示) 内に記憶でき、よって同一構成要素を使用する将来のパイプライン設計の設計及び試験の時間を低減する。即ち、ライブラリから HDL を使用することによって、設計者はスクラッチから先行して具現化された構成要素を再設計する必要性がなく、よって設計者の努力を先行して具現化されていない構成要素の設計に対して、或は、先行して具現化された構成要素の変更に対して集中できる。更には、ライブラリ内にパイプライン回路 80 の多数バージョン或はパイプライン加速器 44 の多数バージョンを画成する HDL を記憶できて、既存の設計の中から精選及び選択できるように為す。

#### 【0051】

図 4 で更に参照されるように、通信インターフェース 82 はメッセージ・ハンドラー 64 (図 3) によって認識されるフォーマットで (ここではバス・インターフェース 91 を介して) データを送受信し、よってピアベクトル・マシン 40 (図 3) の設計及び変更を典型的には補助する。例えば、もしデータ・フォーマットが高速 1/0 フォーマット等の工業規格であれば、ホストプロセッサ 42 及びパイプライン・ユニット 78 の間にカスタムインターフェースを設計する必要がない。更には、パイプライン・ユニット 78 を非バス・インターフェースの代わりにパイプライン・バス 50 を介してホストプロセッサ 42 (図 3) 等の他のピアと通信させることを可能することによって、パイプライン・ユニットが追加或は除去されるたびにスクラッチから非バス・インターフェースを再設計する代わりにパイプライン・バスにそれら (又はそれらを保持する回路カード) を単に接続或は接続解除することによってパイプライン・ユニットの数を変更できる。

#### 【0052】

ハードウェアに組み込まれたパイプライン 74<sub>1</sub>~74<sub>n</sub> は図 3 と連携して先に議論されたように且つ先行して引用された「改善された計算アーキテクチャ、関連システム、並びに、方法」と題された特許文献 2 や「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献 4 におけるように、データに対する各演算を実行し、通信シェル 84 はパイプラインをパイプライン回路 80 の他の構成要素やパイプライン・ユニット 78 の他の回路 (以下に議論されるデータ・メモリ 92 等) にインターフェースする。

#### 【0053】

コントローラ 86 は SYNC 信号や他のピアからの特別なパイプライン・バス通信 (即ち、「事象」) に応じてハードウェアに組み込まれたパイプライン 74<sub>1</sub>~74<sub>n</sub> を同期さ



せ、パイプラインがそれらの各データ演算を実行するシーケンスをモニタし制御する。例えば、ホストプロセッサ42等のピアはSYNC信号を脈動するか、パイプライン・ユニット78に事象をパイプライン・バス50を介して送信して、ピアがデータ・ブロックをパイプライン・ユニットに送信し終えたことを示し、そして、ハードウェアに組み込まれたパイプライン74<sub>1</sub>〜74<sub>n</sub>にこのデータを処理し始めさせる。典型的にはSYNC信号が使用されてタイムクリティカルな演算を同期し、事象が使用されて、非タイムクリティカルな演算を同期する。典型的には、事象はしばしば「ドアベル」と呼称されるデータ無し通信である。しかし事象はデータを含み得て、その場合しばしば「事象メッセージ」と呼称される。SYNC信号及び事象は、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献4に更に議論されている。

10

#### 【0054】

例外マネージャ88はハードウェアに組み込まれたパイプライン74<sub>1</sub>〜74<sub>n</sub>、通信インターフェース82、通信シフル84、コントローラ86、並びに、バス・インターフェース91（もしあれば）の状況をモニタし、ホストプロセッサ42（図3）に例外を報告する。例えば、もし通信インターフェース82におけるバッファがオーバーフローすれば、例外マネージャ88はこれをホストプロセッサ42に報告する。例外マネージャはその例外を生んだ問題を修正するか或はその修正を試みることも可能である。例えば、オーバーフローしているバッファに対して例外マネージャ88は、直接的或は以下に議論されるようなコンフィギュレーション・マネージャ90を介して、そのバッファのサイズを増大し得る。

20

#### 【0055】

コンフィギュレーション・マネージャ90はハードウェアに組み込まれたパイプライン74<sub>1</sub>〜74<sub>n</sub>、通信インターフェース82、通信シフル84、コントローラ86、例外マネージャ88、並びに、インターフェース91（もしあれば）のソフト・コンフィギュレーションを、ホストプロセッサ42（図3）からのソフト・コンフィギュレーション・データに応じて設定し、これは先に引用された「改善された計算アーキテクチャ、関連システム、並びに、方法」と題された特許文献2に議論され、ハード・コンフィギュレーションはパイプライン回路80のトランジスタ及び回路ブロックのレベル上における実際のトポロジーを示し、ソフト・コンフィギュレーションはハード構成された構成要素の物理的パラメータ（例えば、データ幅、テーブル・サイズ）を示す。即ち、ソフト・コンフィギュレーション・データはプロセッサ（図4に不図示）のレジスタにロードされ得るプロセッサの動作モード（例えば、バースト・メモリ・モード）を設定するデータと同様である。例えばホストプロセッサ42は、コンフィギュレーション・マネージャ90に通信インターフェース82におけるデータ及び事象キューの数及び各優先レベルを設定させるソフト・コンフィギュレーション・データを送信し得る。例外マネージャ88は、コンフィギュレーション・マネージャ90に、例えば、通信インターフェース82におけるオーバーフローしているバッファのサイズを増大させるソフト・コンフィギュレーション・データをも送信し得る。

30

#### 【0056】

工業規格バス・インターフェース91は、通信インターフェース82から幾つかのインターフェース回路を効率的にオフロードすることによって、通信インターフェース82のサイズ及び複雑性を低減する従来のバス・インターフェース回路である。それ故に、もしパイプライン・バス50或はルータ61（図3）のパラメータを変更することを望めば、インターフェース91を変更するだけでよく、通信インターフェース82を変更する必要がない。代替的には、パイプライン回路80の外部であるIC（不図示）内にインターフェース91を配置し得る。パイプライン回路80からインターフェース91をオフロードすることは、例えばハードウェアに組み込まれたパイプライン74<sub>1</sub>〜74<sub>n</sub>及びコントローラ86の用途のパイプライン回路上のリソースを解放する。或は、先に議論されたように、バス・インターフェース91は通信インターフェース82の一部であり得る。

40

50

## 【0057】

図4で更に参照されるように、パイプライン回路80に加えて、加速器44のパイプライン・ユニット78はデータ・メモリ92を含み、そしてもしパイプライン回路がPLICであれば、ファームウェア・メモリ52を含む。

## 【0058】

データ・メモリ92は、データがホストプロセッサ42（図3）等の別のピアとハードウェアに組み込まれたパイプライン74<sub>1</sub>〜74<sub>n</sub>との間を流れる時にそのデータをバッファするものであり、ハードウェアに組み込まれたパイプラインに対する作業メモリでもある。通信インターフェース82はデータ・メモリ92を（通信バス94及びもしあれば工業規格インターフェース91を介して）パイプライン・バス50とインターフェースし、通信シェル84はそのデータ・メモリをハードウェア・パイプライン74<sub>1</sub>〜74<sub>n</sub>とインターフェースする。

## 【0059】

データ・メモリ92（或はパイプライン・ユニット78の他の部分）はパイプライン・ユニットのプロファイルをも記憶し得る。そのプロファイルはホストプロセッサ42（図3）に対してパイプライン・ユニット78を十分に記述して、それ自体、パイプライン・ユニット、ピア・ベクトル・マシン40（図3）の他のピアを相互通信用に適切に構成する。例えば、プロファイルはパイプライン・ユニット78が履行することができるデータ演算や通信プロトコルを識別できる。結果として、ピア・ベクトル・マシン40の初期化中にプロファイルを読むことによって、ホストプロセッサ42はメッセージ・ハンドラー64（図3）を適切に構成できて、パイプライン・ユニット78との通信を為す。この技術は「プラグ・アンド・プレイ」技術と類似しており、それによってコンピュータはそれ自体を構成できて、ディスク・ドライブ等の新しくインストールされた周辺機器と通信する。ホストプロセッサ42及びパイプライン・ユニット78のコンフィギュレーションは、先行して引用された「改善された計算アーキテクチャ、関連システム、並びに、方法」と題された特許文献2や、「プログラマブル回路、関連計算マシン、並びに、方法」と題された特許文献5に更に議論されている。

## 【0060】

パイプライン回路80がPLICである図3と連携されて先に議論されたように、ファームウェア・メモリ52はパイプライン回路のハード・コンフィギュレーションを設定するファームウェアを記憶する。このメモリ52はファームウェアを加速器44の構成中にパイプライン回路80にロードし、加速器44の構成中或はその後に通信インターフェース82を介してホストプロセッサ42（図3）から変更されたファームウェアを受信し得る。ファームウェアのローディング及び受信は、先行して引用された「改善された計算アーキテクチャ、関連システム、並びに、方法」と題された特許文献2、「改善された計算アーキテクチャを有する計算マシン、関連システム、並びに、方法」と題された特許文献3、「プログラマブル回路、関連計算マシン、並びに、方法」と題された特許文献5に更に議論されている。

## 【0061】

図4で更に参照されるように、パイプライン・ユニット78は回路ボード或はカード98を含み得て、その上にパイプライン回路80、データ・メモリ92、並びに、ファームウェア・メモリ52が配置されている。この回路ボード98は、ドーターカードがパーソナルコンピュータ（不図示）のマザーボードのスロット内にプラグインされ得ることと非常によく似て、パイプライン・バス・コネクタ（不図示）内にプラグインされ得る。図示されていないが、パイプライン・ユニット78は従来のICや電力調整器及び電力シーケンサ等の構成要素を含み得て、これらIC/構成要素も公知であるようにカード98上に配置され得る。

## 【0062】

パイプライン・ユニット78の構造及び動作の更なる詳細は、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題



された特許文献4に議論されている。

【0063】

図5は、本発明の別の実施例に従った図3のパイプライン加速器44のパイプライン・ユニット100のブロック線図である。パイプライン・ユニット100は、そのパイプライン100が多数のパイプライン回路80。ここでは2つのパイプライン回路80a及び80bを含むことを除いて、図4のパイプライン・ユニット78と類似している。パイプライン回路80の数を増大することは、典型的には、ハードウェアに組み込まれたパイプライン74<sub>1</sub>〜74<sub>n</sub>の数nの増大、よってパイプライン・ユニット78と比較してのパイプライン・ユニット100の機能に関する増大を可能とする。更には、パイプライン・ユニット100はパイプ回路80aに対するファームウェア・メモリ52aとパイプライン回路80bに対するファームウェア・メモリ52bとを含む。代替的には、パイプライン回路80a及び80bは単一ファームウェア・メモリを共有し得る。

10

【0064】

パイプライン・ユニット100において、サービス構成要素、即ち、通信インターフェース82、コントローラ86、例外マネージャ88、コンフィギュレーション・マネージャ90、並びに、任意選択的な工業規格バス・インターフェース91はパイプライン回路80a上に配置され、パイプライン74<sub>1</sub>〜74<sub>n</sub>及び通信シェル84はパイプライン回路80b上に配置される。サービス構成要素及びパイプライン74<sub>1</sub>〜74<sub>n</sub>を個別のパイプライン回路80a及び80b上に位置決めすることによって、サービス構成要素及びパイプラインが同一パイプライン回路上に位置決めされる場合で可能であるものよりも、より多くの数nのパイプライン及び/或はより複雑なパイプラインを含むことができる。代替的には、パイプライン74<sub>1</sub>〜74<sub>n</sub>をインターフェース82及びコントローラ86にインターフェースする通信シェル84の部分はパイプライン回路80a上に配置され得る。

20

【0065】

パイプライン・ユニット100の構造及び動作の更なる詳細は、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献4に議論されている。

【0066】

図6は、本発明の実施例に従った多数のパイプライン・ユニット78（図4）或は100（図5）を有する図3の加速器44のブロック線図である。説明の簡略化のために、加速器44は多数のパイプライン・ユニット78<sub>1</sub>〜78<sub>n</sub>を有するように議論されるが、該加速器が多数のパイプライン・ユニット100或はユニット78及び100の組み合わせを含み得ることを理解して頂きたい。多数のパイプライン・ユニット78を含むことによって、ただ1つのパイプライン・ユニットを有する加速器と比較して加速器44の機能及び処理能力を増大できる。更には、各パイプライン・ユニット78が典型的には一般的な工業規格インターフェースを有するので、パイプライン・ユニットを追加或は削除することで加速器44を容易に変更できる。

30

【0067】

多数のパイプライン加速器44の1つの具現化例において、工業規格バス・インターフェース91は各パイプライン・ユニット78<sub>1</sub>〜78<sub>n</sub>から省略され、単一の（パイプライン・ユニットに対して）外部のインターフェース91及び通信バス94はパイプライン・ユニットの全てに共通している。単一外部バス・インターフェース91を含むことは、図4と連携されて先に議論されたようにパイプライン回路80（図4）上のリソースを解放する。パイプライン・ユニット78<sub>1</sub>〜78<sub>n</sub>は全て単一回路ボード（図6には不図示）上に配置され得るか、各パイプライン・ユニットは各回路ボード上に配置され得るか、或は、グループ分けされた多数のパイプライン・ユニットが多数の回路ボード上にそれぞれ配置され得る。後者の2つの具現化例において、バス・インターフェース91は回路ボードの内の1つの上に配置される。代替的には、パイプライン・ユニット78<sub>1</sub>〜78<sub>n</sub>は、その各々が図4と連携されて先に議論されたように各工業規格バス・インターフェース91を含み得て、よってその各々がパイプライン・バス50或はルータ61（図3）と直に通

40

50

得し得る。この具現化例において、パイプライン・ユニット78<sub>1</sub>〜78<sub>n</sub>は先に議論されたように単一或は多数の回路ボード上に配置され得る。

#### 【0068】

パイプライン・ユニット78<sub>1</sub>〜78<sub>n</sub>の各々はホストプロセッサ42（図3）や相互のピアである。即ち、各パイプライン・ユニット78は通信バス94を介して他の任意のパイプライン・ユニットと直に通信でき、通信バス94、バス・インターフェース91、ルータ61（もしあれば）、並びに、パイプライン・バス50を介してホストプロセッサ42と通信できる。代替的には、パイプライン・ユニット78<sub>1</sub>〜78<sub>n</sub>の各々は各バス・インターフェース91を含み、各パイプライン・ユニットはルータ61（もしあれば）及びパイプライン・バス50を介してホストプロセッサ42と直に通信できる。

10

#### 【0069】

以下、多数のパイプライン・ユニット加速器44の動作は2つの例で説明される。

#### 【0070】

第1例において、パイプライン・ユニット78<sub>1</sub>はデータをパイプライン78<sub>n</sub>に転送し、それがタイムクリティカル方式でデータを処理し、よって、パイプライン・ユニット78<sub>1</sub>及び78<sub>n</sub>は1つ或はそれ以上のSYNC信号を用いてデータ転送及び処理を同期する。典型的には、SYNC信号はタイムクリティカルな機能をトリガーするには十分に速いが、大きなハードウェア・リソースを必要とし、比較して、典型的には事象はタイムクリティカルな機能をトリガーするには十分に速くないが、著しくより少ないハードウェア・リソースを必要とする。先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献4で議論されたように、SYNC信号はピアからピアに直に経路指定されるので、例えばパイプライン・バス50（図3）や通信バス94をトラバースする事象よりもより迅速に機能をトリガーできる。しかし、それは個別に経路指定されるので、SYNC信号は、パイプライン回路80（図4）の経路指定ライン及びバッファ等の専用回路を必要とする。逆に、事象は既存のデータ転送下部構造（例えば、パイプライン・バス50及び通信バス94）を用いるので、より少ない専用ハードウェア・リソースを必要とする。結果として、設計者は全てであるが殆どのタイムクリティカル機能をトリガーするのに事象を使用しがちである。

20

#### 【0071】

第1として、パイプライン・ユニット78<sub>1</sub>はデータを通信バス94に駆動することによってパイプライン・ユニット78<sub>n</sub>に該データを送信する。典型的には、パイプライン・ユニット78<sub>1</sub>はデータとパイプライン・ユニット78<sub>n</sub>のアドレスを含むヘッダーとを含むメッセージを生成する。もしパイプライン・ユニット78<sub>1</sub>がデータを多数のパイプライン・ユニット78<sub>n</sub>に送信するつもりであれば、2つの方法の内の1つでそれを実行し得る。詳細には、パイプライン・ユニット78<sub>1</sub>は個別メッセージを仕向先パイプライン・ユニット78<sub>n</sub>の各々に順次送信し得て、各メッセージは各仕向先ユニットのアドレスを含むヘッダーを含む。代替的には、パイプライン・ユニット78<sub>1</sub>は単一メッセージ・メモリ内にデータと各仕向先パイプライン・ユニットのアドレスを含むヘッダーとを含ませることによって、仕向先パイプライン・ユニット78<sub>n</sub>の各々にデータを同時に送信し得る。データの送信は、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献4に更に議論されている。

30

40

#### 【0072】

次に、パイプライン・ユニット78<sub>1</sub>はデータを受信する。パイプライン・ユニット78<sub>1</sub>〜78<sub>n</sub>は、各々、共通通信バス94と結合され、各パイプライン・ユニット78<sub>1</sub>〜78<sub>n</sub>はデータの意図された受取人であるか否かを決定する。例えば、各パイプライン・ユニット78<sub>1</sub>〜78<sub>n</sub>はそのアドレスがメッセージのヘッダー内に含まれているかを決定する。この例において、ユニット78<sub>1</sub>〜78<sub>n</sub>は、それらがデータの意図された受取人でないことを決定し、よってそのデータを無視、即ち、そのデータをそれらのデータ・メモリ92（図4）にロードしない。逆に、パイプライン・ユニット78<sub>1</sub>はそれがデータの意図された受取人であることを決定し、よってそのデータをそのデータ・メモリ92に

50

ロードする。データの受信は、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献4に更に議論されている。

#### 【0073】

次いで、パイプライン・ユニット78<sub>j</sub>が受信データを処理する準備を為すと、パイプライン・ユニット78<sub>j</sub>等のピア或は外部装置（不図示）がSYNC信号を脈動して、パイプライン・ユニット78<sub>j</sub>にデータを適時に処理させる。SYNC信号を脈動するピア／装置が、パイプライン・ユニット78<sub>j</sub>が受信データを処理する準備をいつ為したかを決定し得る多数の技術が存在する。例えば、ピア／装置は、パイプライン／ユニット78<sub>j</sub>のデータ送信後、所定時間でSYNC信号を単に脈動し得る。おそらく、この所定時間は、パイプライン・ユニット78<sub>j</sub>にデータを受信させてそれをそのデータ・メモリ92（図4）にロードさせるに十分な長さである。代替的にはパイプライン・ユニット78<sub>j</sub>はSYNC信号を脈動して、ピア／装置に受信データを処理する準備が為されたことを知らせる。

#### 【0074】

次に、脈動されたSYNC信号に応じて、パイプライン・ユニット78<sub>j</sub>は受信データを処理する。パイプライン・ユニットによるデータの処理は、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献4に更に議論されている。

#### 【0075】

次いで、パイプライン・ユニット78<sub>j</sub>がデータの処理を終了すると、ピア、外部装置（不図示）、或は、ユニット78<sub>j</sub>自体は、SYNC信号を脈動し得て、パイプライン・ユニット78<sub>j</sub>により多くのデータを送信するように通知する。

#### 【0076】

第2例において、ホストプロセッサ42（図3）はデータをパイプライン78<sub>j</sub>に転送し、それがそのデータを非タイムクリティカル方式で処理し、よってホストプロセッサ及びパイプライン・ユニット78<sub>j</sub>は1つ或はそれ以上の事象を用いてデータ転送及び処理を先に議論された理由から同期する。

#### 【0077】

第1として、ホストプロセッサ42（図3）はデータをパイプライン・バス50（図3）に駆動することによってパイプライン・ユニット78<sub>j</sub>に該データを送信する。典型的には、ホストプロセッサ42はデータとパイプライン・ユニット78<sub>j</sub>のアドレスを含むヘッダーとを含むメッセージを生成する。もしホストプロセッサ42がそのデータを多数のパイプライン・ユニット78<sub>j</sub>に送信するつもりであれば、第1例と連携されて先に議論された2つの方法の内の1つでそれを実行し得る。

#### 【0078】

次に、パイプライン・ユニット78<sub>j</sub>はパイプライン・バス50（図3）から工業規格バス・インターフェース91及び通信バス94を介してデータを受信する。パイプライン・ユニット78<sub>j</sub>〜78<sub>n</sub>が、各々、共通通信バス94と結合されているので、各パイプライン・ユニットはそれが第1例と連携して先に議論された方式でデータの意図された受取人であるかを決定する。

#### 【0079】

次いで、パイプライン・ユニット78<sub>j</sub>が受信データを処理する準備を為すと、ホストプロセッサ42（図3）等のピア或は外部装置（不図示）はパイプライン・バス50上か或は通信バス94の直上に事象を生成して、パイプライン・ユニット78<sub>j</sub>にそのデータを適時に処理させる。事象を生成するピア／装置が、パイプライン・ユニット78<sub>j</sub>が受信データを処理する準備をいつ為したかを決定し得る多数の技術が存在する。例えば、ピア／装置は、ホストプロセッサ42のデータの送信後、所定時間で事象を単に生成し得る。おそらく、この所定時間は、パイプライン・ユニット78<sub>j</sub>にデータを受信させてそれをそのデータ・メモリ92（図4）にロードさせるに十分な長さである。代替的にはパイ

パイプライン・ユニット 78<sub>j</sub>は事象を生成して、ピアノ装置に受信データを処理する準備が為されたことを知らせる。

#### 【0080】

次に、パイプライン・ユニット 78<sub>j</sub>は事象を受信する。事象の受信は、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献 4 に更に議論されている。

#### 【0081】

次いで、受信事象に応じて、パイプライン・ユニット 78<sub>j</sub>は受信データを処理する。パイプライン・ユニット 78<sub>j</sub>によるデータの処理は、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献 4 に更に議論されている。

10

#### 【0082】

次に、パイプライン・ユニット 78<sub>j</sub>がデータの処理を終了すると、ピアノ装置（不図示）、或は、ユニット 78<sub>j</sub>自体は、事象を生成し得て、ホストプロセッサ 42（図 3）により多くのデータを送信するように通知する。

#### 【0083】

図 6 で更に参照されるように、加速器 44 の代替的具現化例が想定されている。例えば、先に議論された動作の第 1 及び第 2 の例は S Y N C 信号及び事象をそれぞれ排他的に使用するが、加速器 44 は S Y N C 信号及び事象の双方を組み合わせて使用できることが想定されている。更には、他のピアノは各データ・メモリ 92 内へのデータの単なるバルク記憶用に多数のパイプライン・ユニット 78 或は 100 の内の 1 つ或はそれ以上を使用できる。加えて、設計者はホストプロセッサ 42（図 3）を、ホストプロセッサの機能を実行する「ホスト」ピアノと一緒に形成するパイプライン・ユニット 78 或は 100 の内の 1 つ或はそれ以上と置き換え可能である。更には、パイプライン・ユニット 78 或は 100 の 1 つ或はそれ以上は 1 つ或はそれ以上のメッセージ分配ピアノとして作用し得る。例えば、ホストプロセッサ 42 が多数の加入者ピアノへの伝送のためのメッセージを生成することを仮定する。ホストプロセッサ 42 はそのメッセージをメッセージ分配ピアノに送信し、該ピアノがそのメッセージを加入者ピアノの各々に分配する。結果として、ホストプロセッサ 42 ではなくそのメッセージ分配ピアノは、メッセージ分配の負荷を請け負い、よってホストプロセッサによってより多くの時間とリソースとを他のタスクに充てさせることを可能とする。

20

30

#### 【0084】

図 7 は、本発明の別の実施例に従った多数のパイプライン・ユニット 78（図 4）或は 100（図 5）を有する加速器 44（図 3）のブロック線図である。

#### 【0085】

図 7 の加速器 44 は、図 7 の加速器 44 が、パイプライン・ユニット 78<sub>j</sub>—78<sub>n</sub>と、ホストプロセッサ 42（図 3）等の他のピアノやパイプライン・バス 50（図 3）と結合されている装置（不図示）との間でのデータの経路指定に対する通信バス・ルータ 110 を含むことを除いて、図 6 の加速器 44 と同一である。説明の簡略化のため、図 7 の加速器 44 は多数のパイプライン・ユニット 78<sub>j</sub>—78<sub>n</sub>を有するように議論されるが、その加速器は多数のパイプライン・ユニット 100 或はユニット 78 及び 100 の組み合わせを含み得ることを理解して頂きたい。

40

#### 【0086】

通信バス・ルータ 110 は通信バス 94 の各分岐 94<sub>j</sub>—94<sub>n</sub>を介してパイプライン・ユニット 78<sub>j</sub>—78<sub>n</sub>と結合されると共に、バス 112 を介して工業規格バス・インターフェース 91（もしあれば）と結合されている。代替的には、図 6 と連携して先に議論されたように、各パイプライン・ユニット 78<sub>j</sub>—78<sub>n</sub>はボード上に各インターフェース 91 を含み得て、よって外部インターフェース 91 は省略されることができて、ルータ 110 は図 3 のパイプライン・バス 50（或はもしあればルータ 61）と直に結合されている。

50

## 【0087】

ルータ110は、パイプライン・バス50（図3）から各仕向先パイプライン・ユニット或は複数の同ユニット78<sub>1</sub>〜78<sub>n</sub>まで信号を経路指定すると共に、ソース・パイプライン・ユニットから1つ或はそれ以上の仕向先パイプライン・ユニット若しくはパイプライン・バスまで信号を経路指定する。結果として、ルータ110はパイプライン・ユニット78<sub>1</sub>〜78<sub>n</sub>の各々から通信バス94上の信号がそのパイプライン・ユニットに向けて意図されているかを決定する機能をオフロードする。このオフローディングは各パイプライン・ユニット78<sub>1</sub>〜78<sub>n</sub>のパイプライン回路80上のリソースを解放し得て、よって各パイプライン・ユニットの機能に関しての増大を許容し得る。

## 【0088】

図7で更に参照されるように、ルータ110を伴う多数のパイプライン・ユニット加速器44の動作を以下に説明する。この動作は図6の加速器44に対しての先に記載されたものと類似しているので、以下の説明は図6及び図7の加速器間の動作差を強調している。

## 【0089】

第1例において、パイプライン・ユニット78<sub>1</sub>はデータをパイプライン・ユニット78<sub>n</sub>に転送し、それがそのデータをタイムクリティカル方式で処理し、よってパイプライン・ユニット78<sub>1</sub>及び78<sub>n</sub>は1つ或はそれ以上のSYNC信号を用いて、図6の第1例と連携して先に議論されたようにそのデータ転送及び処理を同期する。

## 【0090】

第1として、パイプライン・ユニット78<sub>1</sub>はデータを通信バスの分岐94<sub>1</sub>に駆動することによって該データをパイプライン・ユニット78<sub>n</sub>に送信する。典型的には、パイプライン・ユニット78<sub>1</sub>は、データとパイプライン・ユニット78<sub>n</sub>のアドレスを含むヘッダーとを含むメッセージを生成する。

## 【0091】

次に、ルータ110はデータを受信し、そのデータの仕向先がパイプライン・ユニット78<sub>n</sub>であることを決定し、そのデータを通信バスの分岐94<sub>1</sub>に駆動する。典型的には、ルータ110は、データを含むメッセージのヘッダーを分析して、該ヘッダーから仕向先アドレスを抽出することによってデータの仕向先を決定する。結果として、ルータ110はデータの適切な仕向先を決定するので、パイプライン・ユニット78<sub>n</sub>はルータからのデータをデータの意図された受取人であるかを決定することなしに単に受領できる。代替的には、パイプライン78<sub>n</sub>はそれがデータの意図された受取人であるかを決定し得て、もしそれが意図された受取人でなければ例外を生成する（先行して引用された「改善された計算アーキテクチャ、関連システム、並びに、方法」と題された特許文献2、「改善された計算アーキテクチャを有する計算マシン、関連システム、並びに、方法」と題された特許文献3、「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献4で議論されたように）。パイプライン・ユニット78<sub>n</sub>はこの例外をルータ110、工業規格バス・インターフェース91（もしあれば）、ルータ61（もしあれば）、並びに、パイプライン・バス50（図3）を介してホストプロセッサ42（図3）に送信できる。

## 【0092】

次いで、パイプライン・ユニット78<sub>n</sub>はバス分岐94<sub>1</sub>からデータをロードする。パイプライン・ユニットによるデータのローディングは、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献4で更に議論されている。

## 【0093】

次に、パイプライン・ユニット78<sub>n</sub>が受信データ処理する準備を為すと、パイプライン・ユニット78<sub>1</sub>等のピア或は外部装置（不図示）がSYNC信号を脈動して、図6の第1例と連携されて先に議論されたようにパイプライン・ユニット78<sub>n</sub>にそのデータを適時に処理させる。

## 【0094】

次いで、脈動SYNC信号に応じて、パイプライン・ユニット78<sub>0</sub>は図6の第1例と連携されて先に議論されたように受信データを処理する。

## 【0095】

次に、パイプライン・ユニット78<sub>0</sub>がデータの処理を終了すると、ピア、外部装置（不図示）、或は、ユニット78<sub>0</sub>自体は、SYNC信号を脈動し得て、パイプライン・ユニット78<sub>0</sub>により多くのデータを送信するように通知する。

## 【0096】

第2例において、ホストプロセッサ42（図3）はデータをパイプライン78<sub>0</sub>に転送し、それがそのデータを非タイムクリティカル方式で処理し、よってホストプロセッサ及びパイプライン・ユニット78<sub>0</sub>は1つ或はそれ以上の事象を用いて、図6と連携して先に議論された理由のため、そのデータの転送及び処理を同期する。

## 【0097】

第1として、ホストプロセッサ42（図3）はデータをパイプライン・バス50（図3）に駆動することによってパイプライン・ユニット78<sub>0</sub>に送信する。典型的には、ホストプロセッサ42は、データとパイプライン・ユニット78<sub>0</sub>のアドレスを含むヘッダーとを含むメッセージを生成する。

## 【0098】

次に、ルータ110はそのデータを工業規格バス・インターフェース91（もしあれば）及びバス112を介してパイプライン・バス50（図3）から受信する。

## 【0099】

次いで、ルータ110はそのデータの仕向先がパイプライン・ユニット78<sub>0</sub>であることを決定し、そのデータを通信バスの分岐94<sub>0</sub>に駆動する。典型的には、ルータ110は図7の第1例と連携して先に議論されたようにヘッダーの仕向先を決定する。結果として、ルータ110はデータの適切な仕向先を決定し、パイプライン・ユニット78<sub>0</sub>はルータからデータをデータの意図された受取人であることを決定することなしに単に受領することができる。代替的には、パイプライン78<sub>0</sub>はそれがデータの意図された受取人であるかを決定し得て、もしそれが意図された受取人でなければ例外を生成し（先行して引用された「改善された計算アーキテクチャ、関連システム、並びに、方法」と題された特許文献2、「改善された計算アーキテクチャを有する計算マシン、関連システム、並びに、方法」と題された特許文献3、「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献4で議論されたように）、図6の第2例と連携して先に議論されたように、その例外をホストプロセッサ42（図3）に送信する。

## 【0100】

次に、パイプライン・ユニット78<sub>0</sub>はバス分岐94<sub>0</sub>からデータをロードする。パイプライン・ユニットによるデータのローディングは、先行して引用された「改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献4に更に議論されている。

## 【0101】

次いで、パイプライン・ユニット78<sub>0</sub>が受信データを処理する準備を為すと、ホストプロセッサ42（図3）等のピア或は外部装置（不図示）がパイプライン・バス50上或は通信バスの分岐94<sub>0</sub>、94<sub>1</sub>の内の1つの上に事象を生成して、ユニット78<sub>0</sub>に図6の第2例と連携されて先に議論されたようにそのデータを適時に処理させる。

## 【0102】

次に、ルータ110は事象を受信して、それがパイプライン・ユニット78<sub>0</sub>に対して意図されていることを決定し、その事象をバス分岐94<sub>0</sub>に駆動する。

## 【0103】

次いで、パイプライン・ユニット78<sub>0</sub>はバス分岐94<sub>0</sub>から事象をロードする。パイプライン・ユニット78<sub>0</sub>による事象のローディングは、先行して引用された「改善された計



算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法」と題された特許文献4に更に議論されている。

#### 【0104】

次に、受信事象に応じて、パイプライン・ユニット78<sub>0</sub>は受信データを処理する。

#### 【0105】

次いで、パイプライン・ユニット78<sub>0</sub>がデータの処理を終了すると、ビタ、外部装置（不図示）、或は、ユニット78<sub>0</sub>自体は、事象を生成し得て、ホストプロセッサ42（図3）により多くのデータを送信するように通知する。

#### 【0106】

図7で更に参照されるように、動作の第1及び第2の例がそれぞれSYNC信号及び事象を排他的に使用するが、加速器44がSYNC信号及び事象の双方の組み合わせを使用可能であることが想定されている。

#### 【0107】

図8は、本発明の実施例に従った多数のパイプライン・ユニット78（図4）或は100（図5）の多数グループ120を含む図3の加速器44のブロック線図である。パイプライン・ユニットの多数グループ120を含むことは加速器44の機能を増大し、関連動作を実行するパイプライン・ユニットをグループ分けすることによって設計者に加速器の効率を増大させることを可能としている。説明の簡略化のため、図8の加速器44は多数のパイプライン・ユニット78を有するように議論されるが、該加速器が多数のパイプライン・ユニット100或はユニット78及び100の組み合わせを含み得ることを理解して頂きたい。更には、パイプライン・ユニット78は1葉規格バス・インターフェース91（このインターフェースはこの実施例においては外部にある）を含まないが、別の実施例では含んでもよい。

#### 【0108】

加速器44はパイプライン・ユニット78の6つのグループ120<sub>1</sub>〜120<sub>6</sub>を含み、各グループが3つのパイプライン・ユニットと、パイプライン・ユニットを相互に相互接続すると共に他のパイプライン・ユニット・グループと相互接続する各グループ内通信バス・ルータ110<sub>1</sub>〜110<sub>6</sub>とを有する。加速器44が各々3つのパイプライン・ユニット78から成る6つのグループ120<sub>1</sub>〜120<sub>6</sub>を含むように議論されているが、加速器の他の具現化例は任意数パイプライン・ユニットから成るグループの事実上の任意数を含み得て、グループの全てが同一数のパイプライン・ユニットを有する必要性はない。更には、通信バス・ルータ110<sub>1</sub>〜110<sub>6</sub>は図6の加速器44と連携されて先に議論されたように省略され得る。

#### 【0109】

パイプライン・ユニット・グループ120<sub>1</sub>は3つのパイプライン・ユニット78<sub>1</sub>〜78<sub>3</sub>を含み、それらが図7と連携されて先に議論されたものと類似した方式で通信バスの分岐94<sub>1</sub>、94<sub>2</sub>、並びに、94<sub>3</sub>を介してグループ内通信バス・ルータ110<sub>1</sub>と接続されている。他のグループ120<sub>2</sub>〜120<sub>6</sub>は類似である。

#### 【0110】

グループ120<sub>1</sub>〜120<sub>3</sub>の通信バス・ルータ110<sub>1</sub>〜110<sub>3</sub>は第1レベル・バス126<sub>1</sub>の各分岐124<sub>1</sub>〜124<sub>3</sub>を介して第1レベル・ルータ122<sub>1</sub>と接続されている。ルータ122<sub>1</sub>及びバス126<sub>1</sub>はパイプライン・ユニット78<sub>1</sub>〜78<sub>3</sub>が相互に通信することを可能としている。

#### 【0111】

同様に、通信バス・ルータ110<sub>4</sub>〜110<sub>6</sub>は第1レベル・バス126<sub>2</sub>の各分岐128<sub>1</sub>〜128<sub>3</sub>を介して第1レベル・ルータ122<sub>2</sub>と接続されている。ルータ122<sub>2</sub>及びバス126<sub>2</sub>はパイプライン・ユニット78<sub>4</sub>〜78<sub>6</sub>が相互に通信することを可能としている。

#### 【0112】

第1レベル・ルータ122<sub>1</sub>及び122<sub>2</sub>は第2レベル・バス134の各分岐132<sub>1</sub>〜

132<sub>2</sub>を介して第2レベル・ルータ130と接続されている。ルータ130及びバス134はパイプライン・ユニット78<sub>1</sub>〜78<sub>18</sub>が相互に通信すること及び以下に議論されるように他のピア／装置と通信することを可能としている。

#### 【0113】

パイプライン・バス50及び第2パイプライン・バス136は各工業規格バス・インターフェース91<sub>1</sub>及び91<sub>2</sub>を介して第2レベル・ルータ130と結合されている。第2パイプライン・バス136はホストプロセッサ42（図3）等のピア、或は、パイプライン・バス50と結合されていないハードディスク・ドライブ（不図示）等の周辺機器と結合され得る。更には、バス50及び136の何れか或は双方はネットワーク或はインターネット（何れも不図示）を介してピア或は周辺機器と結合され得て、加速器44がホストプロセッサ42（図3）等の他のピアから遠隔的に位置決め可能である。

10

#### 【0114】

バス138は1つ或はそれ以上のSYNC信号をパイプライン／ユニット78<sub>1</sub>〜78<sub>18</sub>の全てと、ホストプロセッサ42（図3）等の他のピア或は装置（不図示）とに直接接続する。

#### 【0115】

図8で更に参照されるように、動作の1つの例において、パイプライン・ユニット78から成る各グループ120<sub>1</sub>〜120<sub>18</sub>は第2パイプライン・バス136と結合されたソナー・アレイ（不図示）の各センサからのデータを処理する。グループ120<sub>1</sub>のパイプライン・ユニット78<sub>1</sub>〜78<sub>3</sub>は単一ルータ110<sub>1</sub>によって相互接続されているので、それらパイプライン・ユニットは、それらが他のグループ120<sub>2</sub>〜120<sub>18</sub>のパイプライン・ユニット78<sub>4</sub>〜78<sub>18</sub>に対してできるものより迅速に相互に通信することができる。このより高い通信速度は他のグループ120<sub>2</sub>〜120<sub>18</sub>の各々においても存在する。結果として、設計者はデータを頻繁に転送するか或はさもなければそれらの間で通信するパイプライン・ユニットを一緒にグループ分けすることによって、加速器44の処理速度を増大することができる。

20

#### 【0116】

一般に、パイプライン・ユニット78<sub>1</sub>〜78<sub>18</sub>は相互に通信する共に、ホストプロセッサ42（図3）等のピア、及び、図7と連携されて先に議論されたものと類似の方式でバス50及び136と結合された装置と通信する。例えば、バス136と結合されたセンサ（不図示）は工業規格バス・インターフェース91<sub>1</sub>、第2レベル・ルータ130、第1レベル・ルータ122<sub>1</sub>、並びに、グループ内ルータ110<sub>1</sub>を介してパイプライン・ユニット78<sub>1</sub>と通信する。同様に、パイプライン・ユニット78<sub>1</sub>は、ルータ110<sub>1</sub>、122<sub>1</sub>、110<sub>3</sub>を介してパイプライン・ユニット78<sub>3</sub>と通信すると共に、ルータ110<sub>1</sub>、122<sub>1</sub>、130、122<sub>2</sub>、110<sub>4</sub>を介してパイプライン・ユニット78<sub>4</sub>と通信する。

30

#### 【0117】

先行する議論は当業者が本発明を作製し使用することを可能とすべく提示されている。種々実施例への様々な変改は当業者には容易に明かであろうし、ここでの包括的な原則は本発明の精神及び範囲から逸脱することなしに他の実施例及び適用例に適用され得る。よって、本発明は図示された実施例に限定されることが意図されておらず、ここに開示された原理及び特徴と一致した最も広い範囲と一致されるべきものである。

40

#### 【図面の簡単な説明】

#### 【0118】

【図1】図1は、従来の多数プロセッサ・アーキテクチャを有する計算マシンのブロック線図である。

【図2】図2は、従来のハードウェアに組み込まれたパイプラインのブロック線図である。

【図3】図3は、本発明の実施例に従ったピア・ベクトル・アーキテクチャを有する計算マシンのブロック線図である。

50



【図4】図4は、本発明の実施例に従った図3のパイプライン加速器におけるパイプライン・ユニットのブロック線図である。

【図5】図5は、本発明の別の実施例に従った図3のパイプライン加速器におけるパイプライン・ユニットのブロック線図である。

【図6】図6は、本発明の実施例に従った多数パイプライン・ユニットを含む図3のパイプライン加速器のブロック線図である。

【図7】図7は、本発明の別の実施例に従った多数パイプライン・ユニットを含む図3のパイプライン加速器のブロック線図である。

【図8】図8は、本発明の実施例に従った多数パイプライン・ユニットからそれぞれが成る複数のグループを含む図3のパイプライン加速器のブロック線図である。

10

【符号の説明】

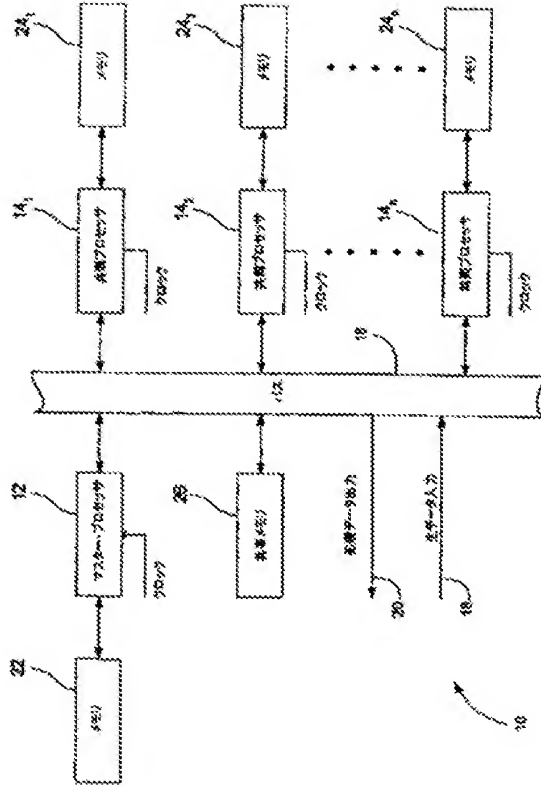
【0119】

10	計算マシン
14	共同プロセッサ
40	ピア-パケットル・マシン
42	ホストプロセッサ
44	パイプライン加速器
46	プロセッサ・メモリ
48	インターフェース・メモリ
50	パイプライン・バス
52	ファームウェア・メモリ
54	生データ入力ポート
58	処理済みデータ出力ポート
61	ルータ
62	処理ユニット
66	処理ユニット・メモリ
68	ハンドラー・メモリ
70	加速器コンフィギュレーション・レジストリ
72	メッセージ・コンフィギュレーション・レジストリ
74	ハードウェアに組み込まれたパイプライン
78	パイプライン・ユニット
80	パイプライン回路
86	パイプライン・コントローラ
88	例外マネージャ
90	コンフィギュレーション・マネージャ
91	工業規格バス・インターフェース
93	通信バス

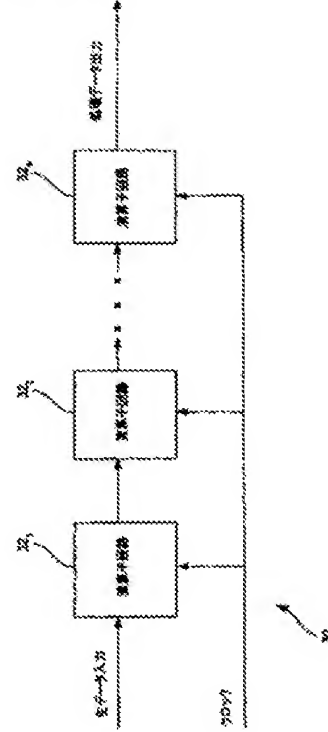
20

30

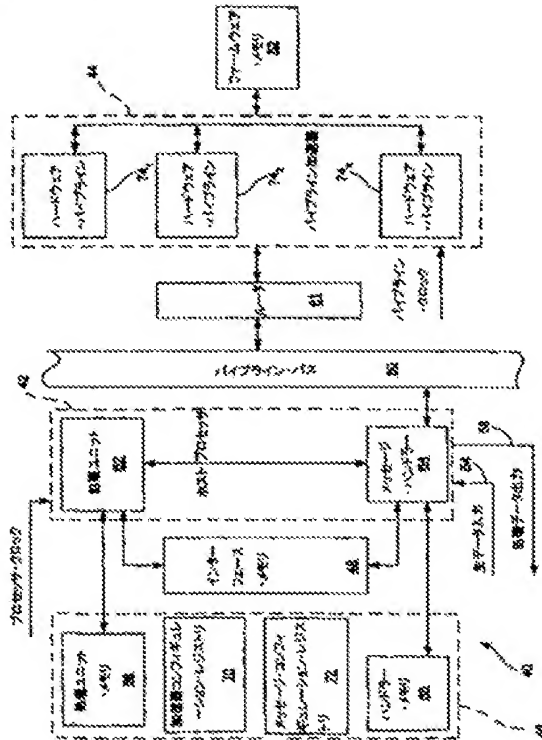
【図 1】



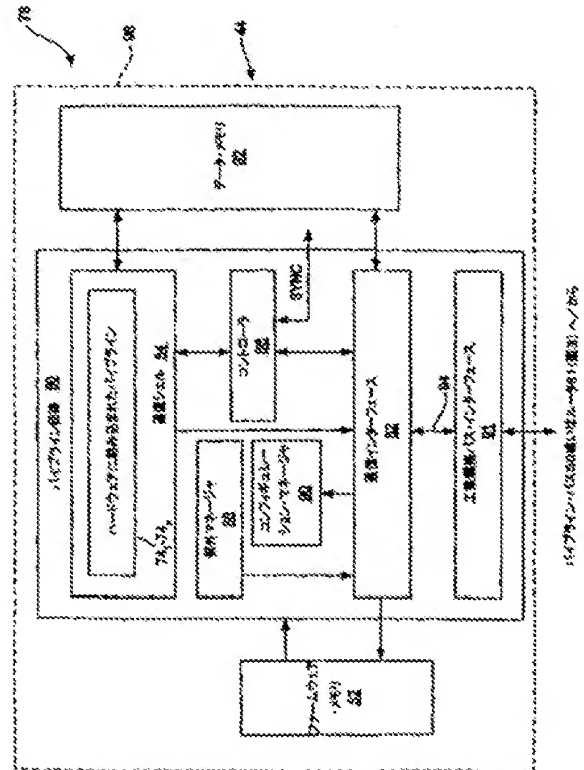
【図 2】



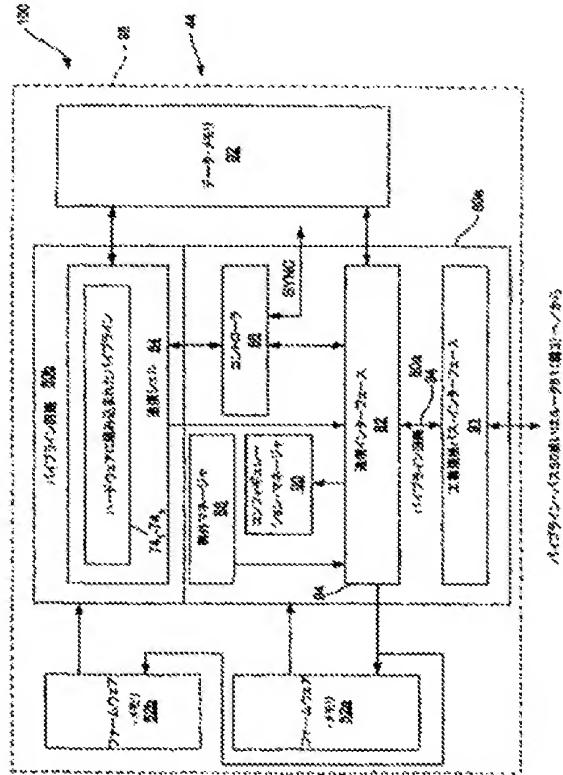
【図 3】



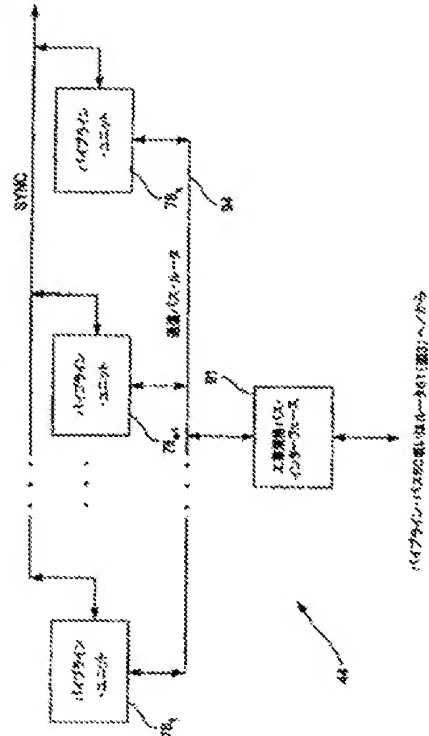
【図 4】



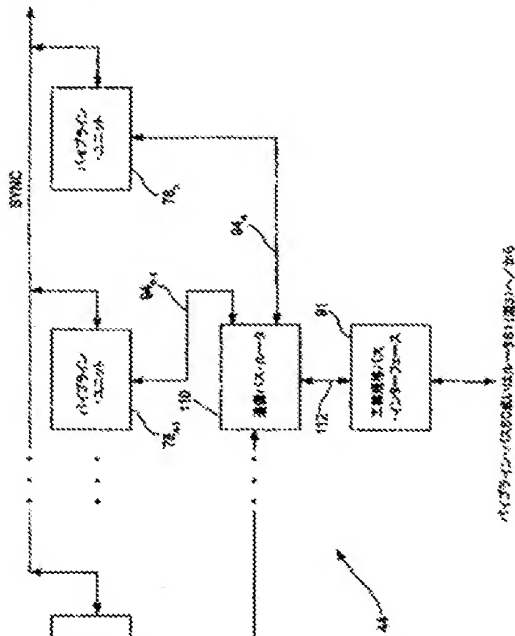
【図5】



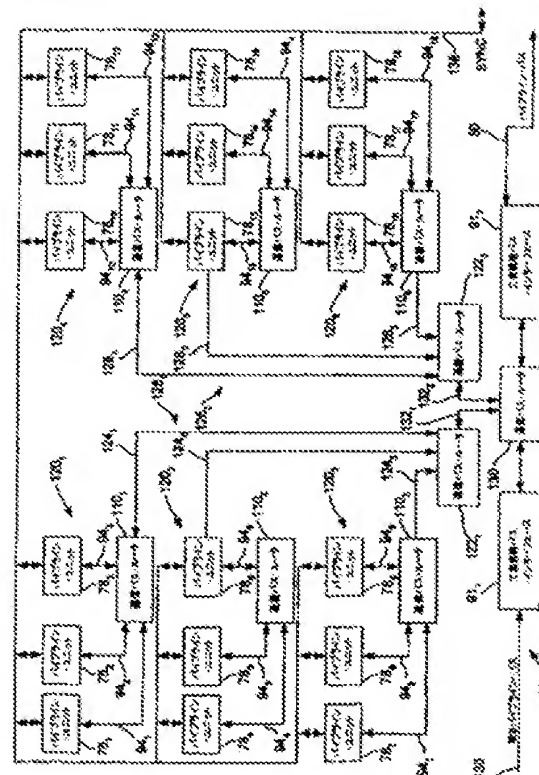
【図6】



【図7】



【図8】



## フロントページの続き

- (31)優先権主張番号 10/684,053  
 (32)優先日 平成15年10月9日(2003.10.9)  
 (33)優先権主張国 米国(US)  
 (31)優先権主張番号 10/684,057  
 (32)優先日 平成15年10月9日(2003.10.9)  
 (33)優先権主張国 米国(US)  
 (31)優先権主張番号 10/684,102  
 (32)優先日 平成15年10月9日(2003.10.9)  
 (33)優先権主張国 米国(US)

(81)指定国 AP(BW, GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), EP(AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GD, GW, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GR, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW

- (74)代理人 100135585  
 弁理士 西尾 務  
 (72)発明者 シュルツ, ケニス, アール,  
 アメリカ合衆国 バージニア州 20112 マナサッス, ゴールデン オータム コート 10  
 506  
 (72)発明者 ラープ, ジョン, ダブリュ,  
 アメリカ合衆国 バージニア州 20110 マナサッス, リバー クレスト ロード 9350  
 (72)発明者 ジャクソン, ラリー,  
 アメリカ合衆国 バージニア州 20112 マナサッス, クレストブロック ドライブ 130  
 93  
 (72)発明者 ジョウンス, マーク,  
 アメリカ合衆国 バージニア州 20120 セントレビル, オークマー プレイス 15342  
 (72)発明者 カーサロ, トロイ,  
 アメリカ合衆国 バージニア州 22701 カルベパー, ケストラル コート 1524